

Towards an Ontology of Performance
Line Pouchard
Computer Science and Mathematics Division
Oak Ridge National Laboratory

Specifying the concepts for an ontology of performance advances PERMIS's project to define metrics of intelligence. The multiple factors briefly mentioned below highlight a few aspects of the complexity and depth of concepts and issues to be considered. It is therefore useful to keep in mind questions that arise when developing an ontology:

- What will be the purpose, use and scope of this ontology? Defining a consensus on terminology, defining machine-readable schemas, a practical and/or theoretical purpose, interoperability between systems, interface between experts in related fields are good examples.
- Are domain specific ontologies advantageous?
- Who will be the users and stakeholders of this ontology? (human communication, computerized interaction, human-machine interaction?)
- How and by whom will it be developed? Which language is chosen for representing the ontologies?

The performance of a distributed system results from trade-offs between properly weighed factors that include computational costs and communication overhead on the one hand, and computational and communication benefits on the other. Computational benefits include the number of transactions per seconds or milliseconds, the throughput of Input/Output of the system as a whole, and its response time.ⁱ Computational costs depend on the load of each component, e.g. number and complexity of processes utilized in performing the task at hand, and the consumption of computational resources. Communication costs may depend on the number of components, their model of interaction and the general architecture of a system.

Computational costs and benefits depend on a compromise between amount and type of resources consumed by the system. For instance, in large distributed systems (order of dozen of nodes distributed across the country) analyzing data-intensive scientific data (order of

petabytes) such as the type of applications for which Grid computingⁱⁱ is designed, the computational costs also vary depending on the type of architecture. The type of architecture here means distributed data and centralized processing, versus distributed data and distributed processing. If data needs to be shipped to a central very powerful computer (super-computer of the type pioneered by Cray research)ⁱⁱⁱ, as current implementation of Grid applications require, the load on computational resources will be large and so will be computational costs. The load in term will affect response time that depends on hardware performance as well as the algorithms driving the hardware.

In the case of distributed data and distributed processing, data is no longer shipped to a central location. Instead, a software code or a software component is moved to a remote computational resource close to data storage.

Communication costs may depend on the number of components interacting with each other, the availability and cost of network bandwidth, and the architecture of communication. Architecture of communication here may mean the protocol of communication^{iv} (for example contract-net protocol in multi-agent systems, asynchronous communication such as message passing in parallel systems). Communication overhead may also include factors related to the structure of a system, whether a hierarchical structure or a web-based structure are used. In the hierarchical structure, children interact with its parents and children; in a web-based structure, any component may a priori interact with any other.

Affecting performance, one also finds factors related to the individual performance of a component independently of the performance of the system as a whole. The knowledge base of a component and the complexity of its rules for reasoning upon input are such factors. Thus several levels of measuring performance in a system may be envisioned.